STRATUM: A Methodology for Designing Heuristic Agent Negotiation Strategies

Iyad Rahwan,[1,2] Liz Sonenberg,[3] Nicholas R. Jennings,[4] Peter McBurney[5]

[1] (Correspondence Address)
Institute of Informatics
The British University in Dubai
P.O.Box 502216, Dubai, UAE
irahwan@acm.org

[3] Department of Information Systems
The University of Melbourne
Parkville 3010, Australia
l.sonenberg@unimelb.edu.au

[2] (Fellow) School of Informatics
The University of Edinburgh
Edinburgh EH8 9LE, UK

[4] School of Electronics & Comp. Sci.
University of Southampton
Southampton, SO17 1BJ, UK
nrj@ecs.soton.ac.uk

[5] Dept. of Computer Science
University of Liverpool
Liverpool, L69 7ZF, UK
p.j.mcburney@csc.liv.ac.uk

September 8, 2006

Abstract

Automated negotiation is a powerful (and sometimes essential) means for allocating resources among self-interested autonomous software agents. A key problem in building negotiating agents is the design of the *negotiation strategy*, which is used by an agent to decide its negotiation behaviour. In complex domains, there is no single, obvious optimal strategy. This has le4d to much work on designing heuristic strategies, where agent designers usually rely on intuition and experience. In this paper, we introduce STRATUM, a methodology for designing strategies for negotiating agents. The methodology provides a disciplined approach to analysing the negotiation environment and designing strategies in light of agent capabilities, and acts as a bridge between theoretical studies of automated negotiation and the software engineering of negotiation applications. We illustrate the application of the methodology by characterising some strategies for the Trading Agent Competition and for argumentation-based negotiation.

1   Introduction

Computer systems are increasingly being viewed in terms of multiple, interacting *autonomous agents*. This is because the multi-agent paradigm offers a powerful set of metaphors, concepts and techniques for conceptualising, designing, implementing and verifying complex distributed systems (Jennings, 2001). In this context, an agent is viewed as an encapsulated computer system that is situated in an environment and is capable of flexible, autonomous action in order to meet its design objectives (Wooldridge, 2002). Building upon this, applications of agent technology have ranged from electronic trading and distributed business process management, to air-traffic and industrial control, to health care and patient monitoring, to gaming and interactive entertainment (Jennings and Wooldridge, 1998; Parunak, 1999; Luck et al., 2003).

In more detail, in a multi-agent system (MAS), agents need to interact. For example, when agents are *cooperative* (e.g. coordinating robots in a disaster rescue application), they need to communicate in order to obtain a shared view of the environment and to coordinate their activities effectively (Tambe et al., 2005). In applications where agents are *self-interested* (e.g. in a trading environment), they also need to interact in order to negotiate and agree on resource exchanges. In many such applications, communication is essential due to the lack of central coordination or resource allocation mechanisms.

We offer the following definition of *negotiation*, adapted from work on the philosophy of argumentation (Walton and Krabbe, 1995):

> *Negotiation is a form of interaction in which a group of agents, with conflicting interests and a desire to cooperate, try to reach a mutually acceptable agreement on the division of scarce resources.*

The use of the word "resources" in the definition above is to be taken in the broadest possible sense. Thus, resources can be commodities, services, time, money etc. In short, anything that is needed to achieve

something. Resources are "scarce" since competing claims over them cannot be fully simultaneously satisfied. In a multi-agent system context, the challenge of automated negotiation is to design mechanisms for allocating resources among software processes representing self-interested parties, be these parties human individuals, organisations, or other software processes.

To date, various interaction and decision mechanisms for automated negotiation have been proposed and studied. These include: game-theoretic analysis (Kraus, 2001; Rosenschein and Zlotkin, 1994; Sandholm, 2002); heuristic-based approaches (Faratin, 2000; Fatima et al., 2002; Kowalczyk and Bui, 2001); and argumentation-based approaches (Kraus et al., 1998; Parsons et al., 1998; Sierra et al., 1998).

A central feature of all these negotiation mechanisms is that agents have some choice of what they may utter, and possibly when they may make utterances. Open-cry auction participants, for example, choose both the content of their utterances (within the constraints of the particular auction protocol) and the timing of their utterances; participants in a sealed-bid, single-round auction may only choose the content. In argumentation-based approaches, participants have still greater freedom in their choice of the content (covering issues such as threats, promises, appeals, etc.) and timing of utterances.

Given this range of options, agent designers therefore face an important question:

What should an agent say, and when, in a particular negotiation interaction?

To answer this question, we turn to the realm of *negotiation strategies* which are here defined as rules or algorithms which provide an answer to this question.

Now for certain types of negotiation frameworks, game theory can be used to characterise optimal strategies and predict the outcomes of a game (Binmore and Vulkan, 1997). However, it has been argued that game-theoretic approaches are insufficient for designing strategies in complex domains (Jennings et al., 2001). This leaves a gap to be filled, as there are no structured approaches to designing strategies for such complex domains. In fact, most strategies in such frameworks have thus far been designed by resorting more or less to intuition and experience. To rectify this, we present a methodology which guides agent designers in the design and selection of strategies for agents engaged in negotiation interactions. The methodology provides a disciplined approach to analysing the negotiation environment and designing strategies in light of agent capabilities.

The paper advances the state of the art in automated negotiation in two main ways. Firstly, the paper presents the first structured methodology for guiding agent designers in designing and selecting strategies in complex, open negotiation domains. Here we exemplify such domains by considering the Trading Agent Competition and argumentation-based negotiation. Secondly, the methodology presents a step towards understanding the common underlying principles of negotiation strategies in various complex domains. This has the potential to enable agent designers to reuse strategies or strategy components across multiple domains and select between strategies based on the characteristics of a given domain.

Hence, our methodology can act as a bridge between the theoretical and simulation-based studies of automated negotiation on one hand, and the software engineering of practical applications of automated negotiation on the other.

The rest of the paper is organised as follows. In the next section, we motivate the need for a strategy design methodology. In Section 3, we present the STRATUM methodology for strategy design. Then, in Section 4, we demonstrate how the methodology can be applied by considering two case studies: one from the Trading Agent Competition (Eriksson and Janson, 2002); the other from argumentation-based negotiation (Rahwan et al., 2003). We conclude the paper in Section 5.

2 Why a Methodology is Needed

In this section, we motivate the need for a strategy design methodology. In the next subsection, we demonstrate that the analytical approach of game-theory is not suitable in certain domains. Then, we also demonstrate why agent-oriented software engineering methodologies fall short of providing structure to the strategy design process. The last subsection scopes the contribution of the paper in light of these observations.

2.1 Relation to Automated Negotiation Approaches

Researchers in the area of MAS initially used the tools of classical game theory to design strategies for negotiating agents (Rosenschein and Zlotkin, 1994). In game-theoretic analysis, researchers typically attempt to determine the optimal strategy by analysing the interaction as a game between participants and seeking its equilibrium (Harsanyi, 1956; von Stengel, 2002). The strategy determined by these methods is typically shown to be optimal for a participant, given the game rules, the assumed payoffs and preferences of participants, and assuming that agents have common knowledge that they are all rational[1] and that participants have no knowledge of one another not provided by introspection. In such cases, it is possible to use game-theoretic techniques in order to design resource allocation mechanisms that force agents to follow prescribed optimal strategies (Dash et al., 2003).

However, it quickly turned out that the assumptions of game theory are often not satisfied in multi-agent systems (Jennings et al., 2001). The results of classical game theory (e.g. about the optimal strategies) are only valid under the assumptions of perfect rationality.[2] However, in real applications, agents may be resource-constrained, malicious or whimsical, or simply badly-coded, so that participant

---

[1]Rational in the sense that they try to maximise their expected utilities, that their preferences satisfy certain axiomatic conditions, and that their decisions assume that other agents are also rational.

[2]Some work has be done on calculating equilibrium strategies while taking into account the cost of computation (Larson and Sandholm, 2001). However, this approach has been applied to a particular, relatively simple bilateral bargaining setting, and it is not clear yet how this approach generalises to more complex negotiations (e.g. involving multiple concurrent negotiations).

behaviour may not conform to the assumptions of economic rationality. In such cases, researchers have turned to heuristic methods. Heuristics are rules of thumb that produce "good enough" outcomes, and are mainly based on empirical testing and evaluation. In heuristic-based frameworks, strategies have been proposed which are based on, for example, the underlying utility model, decay functions of factors such as utility and time (Faratin, 2000; Kraus, 2001), or fuzzy modelling of the environment (He et al., 2003). Very specific bargaining strategies have been analyzed, for example, by investigating their optimality (Fatima et al., 2001), their performance in multiple negotiation rounds (Faratin, 2000), or the resulting social welfare (Harsanyi, 1956).

However, there is no "methodology" for guiding designers of strategies in heuristic-based frameworks. So, given a new type of negotiation domain, such as the travel services market of the Trading Agent Competition (TAC) (TAC, 2003), there is little guidance on how one might go about designing negotiation strategies.

Another type of framework which challenge game theoretic approaches is what is becoming known as the argument-based negotiation approach (Rahwan et al., 2003). In these frameworks, agents can exchange, in addition to proposals and indications of their acceptance or rejection, other meta-information about proposals, such as the reasons for proposals, and the reasons for accepting or rejecting them. Here, strategies were also proposed on an ad hoc basis (Sierra et al., 1998; Sadri et al., 2001) possibly inspired by the psychology of human persuasion (Kraus et al., 1998; Ramchurn et al., 2003). Thus, there has been little work that investigates generic principles underlying these heuristics.

While the game-theoretic approach is well understood, we are motivated by the lack of methodology for guiding designers of strategies in heuristic-based or argumentation-based negotiation encounters. It would be of value, therefore, to provide a methodology that gives generic guidelines for strategy designers operating under such frameworks. To this end, this paper takes the first step towards characterising a methodology for designing strategies under complex protocols and relaxed assumptions. Thus, given a specific heuristic or argumentation-based encounter described in terms of interaction rules and a set of agents participating in negotiation, the methodology would assist agent designers in distilling essential features of the framework and using these to incrementally design sophisticated strategies.

2.2   Relation to AOSE Methodologies

In recent years, there has been an increase in research on Agent-Oriented Software Engineering (AOSE) methodologies. A number of AOSE methodologies have been proposed for guiding the design and construction of multi-agent systems, such as GAIA (Zambonelli et al., 2003; Wooldridge et al., 2000), Tropos (Bresciani et al., 2004), ROADMAP (Juan et al., 2002), Prometheus (Padgham and Winikoff, 2004) OperA (Dignum, 2004) and AUML (Bauer et al., 2001). One might reasonably ask whether AOSE methodologies could provide the answer to the problem of strategy design for protocols that cannot be studied using game-theoretic tools.

Now there have been some attempts to use or produce AOSE methodologies to engineer negotiation protocols and agents. For example, Ashri et al. (2003) use the SMART agent construction model (d'Inverno and Luck, 2001) to specify negotiating agent architectures. Fritschi and Dorer (2002) use an AOSE methodology to construct an agent that participated in the Trading Agent Competition. Kungas and Rao (2004) present a methodology for agent-based Web service discovery and automated composition using negotiation. Dumas et al. (2002) propose a formal language for specifying negotiating agent behaviour, which combines State Charts (Harel and Naamad, 1996) and Defeasible Logic Programs (Billington, 1993).

However, AOSE methodologies are typically concerned with the process of capturing domain requirements *before* the agent system exists, and then transforming these requirements into multi-agent system specifications. During this process, system designers can follow the methodology as they describe the environment, agent capabilities, roles, relationship structures, interaction protocols, and so on. When it comes to specifying agents, current AOSE methodologies usually assume that agents are specified by the same designer or, if specified by multiple designers, that they are cooperative. In an open system, where agents can be programmed by anyone, and cannot be assumed cooperative, current AOSE methodologies provide no advice on the design of negotiation strategies.

The methodology we present here should *not* be seen as a competitor to existing AOSE methodologies, when they are applied to specifying open negotiation environments. Instead, our methodology *complements* AOSE methodologies since we assume a multi-agent system specification is partially given (implicitly or explicitly) using some language, and we provide guidance to an agent designer in programming strategies which operate *within* the given specification.

2.3   Scope

In this subsection, we define more precisely the scope of this paper. This requires a characterisation of the various components of a negotiation framework. A negotiation framework can be seen to involve the following (Bartolini et al., 2002):

1. a **negotiation locale**, which is a communication platform through which agents interact;

2. a **host** (or **monitor**) that manages the negotiation locale and facilitates the negotiation, making sure participants abide by the protocol rules, pay violation fines etc.;

3. a **negotiation template**, which is essentially a language for describing deals;

4. a set of **negotiation rules**, which include:

    (a) *rules for participant admission*;
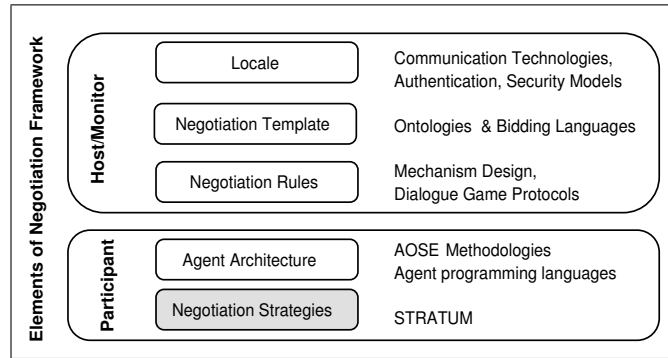
    (b) *rules for checking proposal validity*;

Figure 1: Scope of the STRATUM methodology

(c) *rules for protocol enforcement*, which include rules that specify when agents can make proposals, accept proposals, make other utterances etc.;

(d) *rules for updating status and informing participants*, which specify when agents can access what information and how the overall state of the negotiation (e.g. current highest bid) is updated;

(e) *rules for agreement formation*, which specify when an agreement is reached (e.g. an English auction ends with agreement if an acceptable bid is not exceeded by another bid within a specified time period);

(f) *termination rules*, which specify under what conditions the negotiation terminates (successfully or unsuccessfully);

5. a number of **participants**, with their internal decision-making components (e.g. their architectures, proposal evaluation logic etc.) and their negotiation strategies;

This paper is specifically concerned with the design of the *participants* in a negotiation framework, and in particular with their *strategies*. To this end, Figure 1 depicts the scope of our methodology, which has the specific purpose of helping designers of participating agents specify their agents' negotiation strategies. We are hence not concerned with guiding software engineers in designing the rules of negotiation or specifying languages for describing agreements etc. In fact, we assume a negotiation framework or mechanisms is *given* and specified, formally or informally, in some form. However, since the effectiveness of strategies is highly dependent on the nature of the underlying negotiation framework (rules, templates etc.), the methodology also guides agent strategy designers in distilling framework characteristics that are relevant and useful for strategy design.
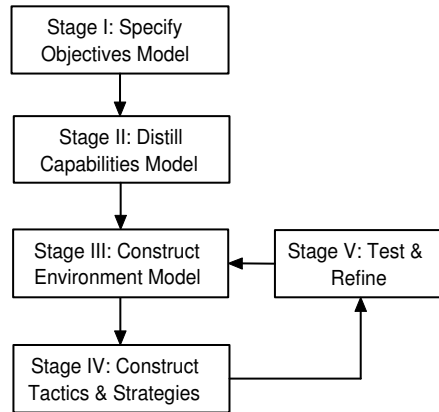
Figure 2: Stages of the methodology

3 A Methodology for Strategy Design

In this section, we introduce the **STRATUM** (for **STRAT**egy via **U**northodox **M**ethodology)[3] methodology for designing agent negotiation strategies.

The main function of the STRATUM methodology is to guide the designer of negotiation strategies from domain and requirements analysis to producing modular high-level specifications of strategies. These specifications should be generic enough to provide flexibility in the underlying implementation details and specific enough to provide thorough and useful guidance for programming strategies.

Typically, methodologies for software construction comprise a number of stages. Each stage may result in one or more *models* that describe various aspects of the problem or the system using some informal or formal abstraction or language. STRATUM consists of the stages described in Figure 2. The stages arose from our earlier work on characterising strategic factors in negotiation (Rahwan et al., 2003). These stages should not be understood as a strictly sequential process. The arrows represent the overall direction of logical dependency between the models, but the designer may do the development by iterative refinement across stages. In the next sub-sections, we discuss each of these stages in more detail.

3.1 Stage I: Specify Objectives Model

The Objectives Model (OM) specifies the objectives of the agent. This may be as simple as "maximise own expected utility," or involve a more complex specification of the types of goals the agent needs to achieve in the world (e.g. to increase the market share and minimise negotiation time), the safety conditions it has to maintain (e.g. maintain minimum budget deficit), the constraints imposed on these

---

[3]The name illustrates that this methodology does not follow the traditional game-theoretic approach. The word *stratum* is the singular of *strata* (a set of layers) and refers to the compositional construction of strategies from simple tactics and capabilities.

objectives (e.g. never concede on quality) and so on. The OM may be expressed using natural language statements, but may also be specified formally, for example using some objective or utility function.

We assume that we are dealing only with purposeful agents. Hence, each party to a negotiation may be assumed to have some objectives which lead it to enter into the negotiation interaction. These objectives may be at the highest level of an agent's stack of goals or they may not be, in which case they may support some other, even higher, goals. Entering into a particular negotiation interaction over certain resources with particular agents at a particular time, will, the agent believes, assist it in seeking to achieve these goals.

An agent's negotiation objectives may be a particular agreement to divide the scarce resources under discussion. Such an outcome, however, is not the only objective an agent may have. Indeed, an agent may enter into an interaction with no intention of seeking a division of the resources in question, but merely to confuse, distract, or otherwise delude the other participants, or even non-participants. An agent may also engage in a negotiation interaction to acquire information about a new domain, as when potential house-buyers participate in auctions in order, not to purchase a house, but to educate themselves about prevailing house prices, or even about the auction process itself.

Agents may even enter into a negotiation interaction with one counterpart in order to have a stronger negotiation position relative to another counterpart in a separate interaction; business-to-business negotiations often involve such parallel, competitive negotiations (Lilien et al., 1992, Chapter 3). These objectives are all valid — and, by any definition, rational — objectives from a negotiation interaction. Their wisdom, feasibility, or ethical content, though, are separate issues.

Note that the agent designer may not know precisely, at design time, in what encounters and with what counterparts the agent will be engaged. However, since agents are assumed to be "goal directed" or "purposeful," it should be possible to spell out their overall objective. This overall objective may be realised in different ways depending on the type of encounter. Such detail is specified in stages III and IV, which are described below.

3.2   Stage II: Distill Capabilities Model

A key influencing factor on strategy design is the interaction capabilities of the agent negotiator — what the agent is capable of doing in the interaction. These capabilities constitute the agent's Capabilities Model (CM). These interaction capabilities are mainly characterised by the interaction protocol the agents use. We assume that the protocol is already given by the negotiation host designer (e.g. specification of auction rules or communication language). Two main types of capabilities may exist:

1. **Dialogical abilities:** These specify what the agent is capable of uttering to other agents. Such capabilities may be specified in terms of the communication language used and the protocol rules that govern the use of this language. Capabilities may also be constrained through some system of value, such as a system for tracking agents' commitments to check their consistency (Maudet

and Chaib-draa, 2003), or a system of reputation or trust that removes agents that behave in a deceptive or misleading manner.

2. **Relevant non-dialogical abilities:** An agent may have non-dialogical capabilities which may impact its dialogical abilities. Consider, for example, an agent who wants to promise another agent to deliver a fuel tank at a certain time. If the mechanism given prohibits agents from lying or decommitting on their promises, then the agent must actually have access to a fuel tank and be capable of moving it to the specified address at the specified time. In other words, the agent can only make the promise (a dialogical ability) if it is capable of fulfilling the promise (a non-dialogical ability).

Our interest here is in the dialogical abilities, and implicitly in the non-dialogical abilities – but only insofar as the latter contribute to the former. So, for example, an agent's ability to inflict physical damage on another agent is only relevant to negotiation if this ability can be used to create dialogical threats against that agent, such as "if you do not accept my proposal, I will hurt you." Finally, we stress that here, we are not interested in the specifics of the agent's *internal* capabilities (e.g. its ability to evaluate an offer or generate preferences), but only in the dialogical manifestations of such abilities (e.g. the ability to communicate desires to other preferences).

At the bottom level, an agent engaged in a negotiation interaction must be able to make utterances which are legal according to the rules of the protocol. Above this level are some higher-order capabilities, which may, depending on the specific protocol, require utterance of a sequence of locutions to be effected. Typical types of dialogical capabilities needed in a negotiation encounter are listed in Table 1. This set of capabilities was compiled based on observations of the automated negotiation literature (Jennings et al., 2001) as well as literature on human negotiation (Young, 2001; Lewicki et al., 2003). A specific protocol may enable only a subset of these capabilities. Moreover, a specific protocol may enable only certain sub-types of these capabilities. For example, a protocol may enable an agent to provide information proactively (capability C4) about its preferences, but not its beliefs. In any case, the agent designer must have a clear picture of the agent's dialogical capabilities.

Of course, an agent may be said to also have capabilities which are complex combinations of these. For example, the ability to prevaricate (Dunne, 2003) may be constructed from abilities to: request irrelevant information; provide irrelevant, misleading or confusing information; or repeat previous questions or statements. Such capabilities may constitute negotiation *tactics* or *strategies* and will be discussed in section 3.4 below.

The CM is specified by the agent designer as a list of capabilities, which is a sub-set of the list given in Table 1. In addition, the designer could provide a detailed description of how each capability can be performed. This may include, for example, the protocol utterances that facilitate each capability, or its enabling non-dialogical capabilities.

	Capability	**Explanation**
C1	*Make proposals*	Proposing potential deals to the counterpart.
C2	*Accept proposals*	–
C3	*Reject proposals*	–
C4	*Present information proactively*	An agent may present information in order to influence a counterparty's beliefs, preferences, intentions etc.
C5	*Seek information from a counterpart*	Participants may have varying abilities to extract information from one another, for example, due to differing levels of authority in a social structure.
C6	*Provide information reactively*	Agents may have differing capabilities to provide information to one another; some agents may not be able to lie, or to answer evasively, for example.
C7	*Seek to exert pressure on counterpart*	An agent might be able to threaten or reward other participants for accepting certain deals (Kraus et al., 1998; Sierra et al., 1998), e.g. using authority.
C8	*Retract Commitments*	Agents may have ability to retract commitments or proposals they have made previously. Retraction has been studied, for example, in argumentation theory (Walton and Krabbe, 1995) and in bargaining (Sandholm and Lesser, 2002).
C9	*Withdraw*	Truly autonomous agents should have the ability to withdraw from any interaction at any stage. Agents may also have the ability to threaten to withdraw.
C10	*Re-enter*	Some auction protocols allow agents to withdraw form auctions and re-enter them later.
C11	*Do nothing*	Be passive and wait until conditions change; E.g. until market prices go down, or until counterpart concedes because of his/her time constraints.

Table 1: Common Types of Dialogical Capabilities in a Negotiation Encounter

When identifying the agent's capabilities, the designer also needs to take account of constraints on the exercise of any potential capabilities. Such constraints could include:

1. *Interaction Protocol*: The rules of the negotiation interaction protocol may preclude or require certain utterances or certain types of utterances by agents at particular times in an interaction. The FIPA Agent Communications Language, FIPA ACL, for example, requires agent sincerity: only statements which are believed by an agent may be uttered using the *inform* locution (FIPA, 2001). In principle, such a condition must severely limit the use of FIPA ACL for negotiations.

2. *Values*: By values, we refer to the "criteria" that motivate and constrain the agents' behaviour. Examples of such values include "maximising expected utility," "maximising social welfare," or "avoiding deceptive behaviour." The agent's values may preclude or require certain behaviours and so constrain the potential capabilities of the agent. Young (2001), for example, argues that the strategies of human negotiators are significantly influenced by the value of maintaining their social identity.

3. *Resource Constraints*: Time, memory or processing limitations on an agent may limit its capabilities in a negotiation interaction.

3.3   Stage III: Construct Environment Model

This stage involves providing some representation of the negotiation environment, including the negotiation counterparts, for the sake of reasoning about them. The result constitutes what we refer to as the Environment Model (EM) model. Reasoning about this model, either by the strategy designer at design-time, or by the automated agent at run-time, would provide a basis for the design and selection of appropriate strategies.

The environment model describes one or more of the following:

- **Model of counterpart**: This model captures how the negotiation counterparts behave, the information they may have, or how they are likely to behave. For example, an agent wishing to acquire a resource may attempt to model the negotiation behaviour of the providers of these resources. Providers may be hard, using set prices, or they may be soft, conceding on price after a few bargaining rounds.

- **Model of peer**: This model captures how other negotiators behave in the system. For example, an agent wishing to acquire a resource may attempt to model other competing consumers of that resource. If there are many consumers with aggressive demand, this may give providers extra market power and hence decrease their willingness to concede with other consumers.

- **Model of external conditions**: Instead of modelling specific behaviour of counterparts or peers, an agent may attempt to model the external conditions that affect these behaviours. For example,

Counterparts & Peers
- *Aggressiveness:* How aggressively is the peer or counterpart buying, selling or bidding?
- *Informativeness:* How much does the counterpart or peer know? What information do they have access to?
- *Trustworthiness:* Can peer or counterpart be trusted to fulfill promises, provide truthful information etc.?
External Conditions
- *Rate of change:* Are there patterns that capture change in external conditions, such as resource supply, demand, prices etc.? What factors are these conditions affected by?
- *Competitiveness:* How competitive are other agents? (This can be seen as a generalisation of the aggressiveness attitude to multiple agents).

Table 2: Typical Environment Characteristics

the designer of an agent bidding in an exchange over oil stocks may attempt to consider the expected economic conditions that would affect oil supplies and consequently demand and prices. It is notable that the model of external conditions only captures the agent's *perception* of these conditions, and hence does not necessarily reflect the *actual* conditions.

Modelling other agents is a growing area of research, and techniques range from probabilistic modelling of individual agent beliefs (Milch and Koller, 2000; Gal and Pfeffer, 2003) to qualitative modelling of group mental attitudes (Boella and van der Torre, 2004). Modelling the economic conditions has also begun to receive some attention in the automated negotiation literature. For example, He et al. (2003) enable agents to model the "state of the marketplace," based on observations of supply and demand, and adjust their bidding strategies accordingly using a meta-strategy based on pre-programmed fuzzy rules.

STRATUM does not prescribe a particular method for modelling the environment, as different methods would be more suitable for different types of negotiation frameworks. However, we do provide a list of typical environment *features* that the strategy designer could look for. These features, described in table 2, can be used as a blueprint to guide the selection of the most appropriate environment modelling method.

3.4 Stage IV: Construct Tactics and Strategies

With the objectives identified, capabilities clarified and environment understood, we can proceed to designing actual strategies. This process constitutes the Tactic and Strategy Design Stage. Intuitively, tactics represent detailed low-level patterns of behaviour, while strategies represent more sophisticated negotiation behaviour that makes use of simpler tactics.

3.4.1 Goal Decomposition vs. Tactic Composition

As stated earlier, we assume an agent enters a particular negotiation interaction over particular resources using a particular interaction protocol with particular counterpart agents at a particular time, in order to achieve its negotiation objectives. In order to achieve these objectives, the agent may attempt to achieve some *sub-objectives*, with the belief that such sub-objectives assist in realizing the overall negotiation objectives.[4] Sub-objectives themselves may be further decomposed into lower-level sub-sub-objectives and so on. We can therefore see the process of strategy design in a compositional fashion.[5]

For example, a potential buyer entering into a negotiation with a car-dealer aiming to buy a car may seek to achieve this negotiation objective by realizing each of the following sub-objectives (in sequence):

X. Learning about the alternative models available from the dealer;

Y. Establishing a preference ordering over some or all of these models; and

Z. Getting a cheap price for the most-preferred model.

The buyer might achieve the first sub-objective by posing a series of questions to the car dealer. The second sub-objective may be achieved by introspection, perhaps involving a process of comparison of the expected utilities of different models (Roberts and Lattin, 1991). To achieve the third sub-objective, the buyer may seek to achieve two lower-level objectives:

Z.1 Informing the dealer about an offer made by a competing dealer; and

Z.2 Bargaining with the dealer through an exchange of offers.

Each of these sub-sub-objectives may be achieved directly by making a series of utterances, or through decomposition into further sub-objectives and so on. This process guides the design of tactics and strategies.

Given this, Figure 3 depicts the interleaving processes of *decomposing negotiation objectives* and *composing capabilities*. Objectives are decomposed in a top-down fashion, while capabilities are composed in a bottom-up fashion in order to construct tactics and strategies that achieve these objectives. For example, to achieve the main negotiation objective, the designer may specify that the agent needs to achieve sub-objectives SO-1 through to SO-n. Each of these sub-objectives may need to be further decomposed into sub-sub-objectives, and so on. The same thing takes place, bottom-up, while composing tactics from capabilities. In the Figure, the arrow between the two trapezoids denotes that capability

---

[4]This hierarchical objectives structure has a structural similarity to the landmarks theory of conversation protocols of Kumar et al. (2002). However, our approach concerns only the objectives of an individual agent and not the joint goals of all participants to an interaction.

[5]When describing the negotiation objectives and strategies, we use the terms "objective" and "sub-objective" deliberately in order to distinguish them from the agent's own internal "goals" and "sub-goals".
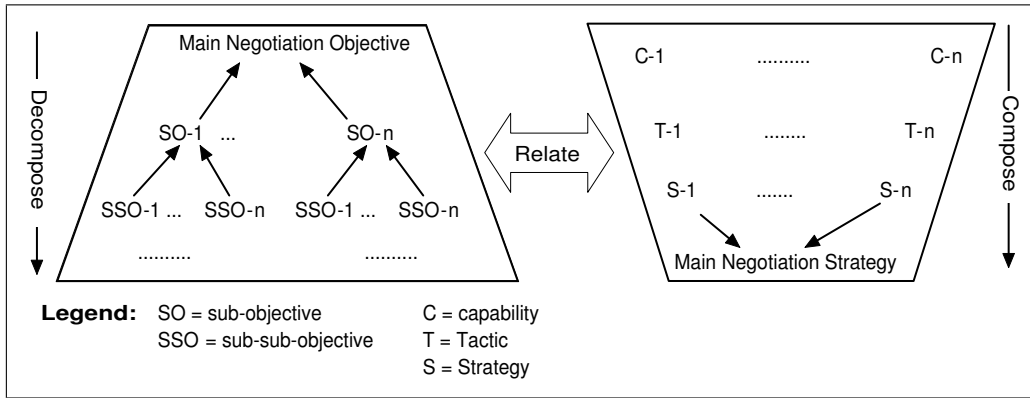
Figure 3: Decomposing objectives top-down; composing capabilities bottom-up

description is informed by the decomposition of objectives and, vice versa, objectives decomposition is informed by the capabilities available.

In the process of objective decomposition and strategy composition, various types of interdependencies may be revealed. First, some objectives may be found to be unachievable because the agent simply does not have the required capabilities. For example, if an agent is not allowed to withdraw from the negotiation, then it cannot threaten to do so. In such cases, alternative sub-objectives should be identified in order to achieve the main negotiation objective.

Similarly, the parallel inspection of objectives and capabilities may reveal new possible objectives and/or tactics/strategies. More specifically, after attempting to map a given sub-objective to a combination of capabilities, the designer may discover that the capabilities can be combined in a way not previously thought of. This combination could achieve a new, alternative sub-objective that proves to be useful.

Objective decomposition and strategy composition may also reveal conflicts among two or more objectives, for example because they require the use of mutually exclusive capabilities. In such cases, the designer should consider trade-offs between different conflicting objectives or seek alternative ways in which conflict could be avoided. The same could be said about discovering conflicts among two or more capabilities, tactics or strategies, which means they should not be used in combination.

At the end of this stage, there must be a correspondence between the objectives hierarchy and the tactics/strategies hierarchy. In other words, each objective or sub-objective, etc., must be achievable through a particular tactic or combination of tactics, and each tactic must have an objective or sub-objective, etc., as its purpose. This correspondence need not be a one-to-one mapping, however, since a single tactic may contribute to more than one objective, and a single objective may require more than one tactic. However, the lack of one-to-one mapping reveals discrepancy in abstraction between the two hierarchies and should be avoided if possible.

3.4.2 Composing Tactics and Strategies

We now list a number of possible low-level *tactics* (or equivalently, low-level sub-objectives), which may contribute to an agent's achievement of its negotiation objectives. To date, no comprehensive list of all possible applicable negotiation tactics is available in the multi-agent literature or in the literature on human negotiation. Therefore, we list those tactics inspired by the multi-agent negotiation literature as well as informal advice to human negotiation participants (Fisher et al., 1991; Young, 2001; Lewicki et al., 2003).

1. **Seek to change a counterpart's beliefs.** One participant in a negotiation may judge it to be in its interests to have other participants believe certain propositions about the beliefs, intentions, preferences, or constraints of the first participant, or about the domain in question. These propositions may be true or false. Providing information to the counterparts may enable an agent to explain the reasons for its beliefs, preferences etc. It has been argued (Fisher et al., 1991; Rahwan et al., 2003) that agreement is more likely in negotiation interactions when participants understand each others' interests (desires, preferences, goals etc.) rather than their current positions.

2. **Gain a better understanding of a counterpart.** An agent may have uncertain or incomplete information about its counterpart. Moreover, counterparts may be seeking to mislead a participant about their beliefs, intentions, preferences, constraints etc., or about the domain. An agent may then seek to gain a better understanding of its counterparts' true mental states or constraints.

3. **Seek to discuss a particular issue.** By moving the interaction towards particular issues, a participant may be able to frame the problem in certain ways and thus influence the mental states of its counterparts. A seller of a particular make of car, for example, may seek to turn the topic of discussions with potential buyers towards attributes on which this make of car scores highly.

4. **Seek to avoid discussion.** For the same reasons, a participant may wish to steer discussion away from particular issues.

5. **Seek fast termination.** An agent with time or processing resource constraints might seek a fast resolution or termination of the negotiation.

6. **Seek to delay.** An agent who believes it has greater time or other resources than other participants may seek to delay resolution of the interaction beyond the perceived resource limits of its counterparty (Dunne, 2003).

7. **Resist a counterpart.** An agent may resist attempts by a counterpart to achieve one of the above tactics.

With these main *tactic types* in mind, the strategy designer can start specifying tactics to be used by the software agent. Specification of these tactics must take into account the actual capabilities of the

Tactic Name	- Tactic 1
Dialogical Objective	Get a cheap price for a preferred car model
Method	- Inform the dealer about an offer made by a competing dealer; - Bargain with the dealer by exchanging offers.
Capabilities & Sub-Tactics Used	- Capability C4 to provide information proactively; - Capability C7 to exert pressure on the car dealer; - Capabilities C1, C2 and C3 to make, accept and reject proposals, respectively.
Rationality Condition	None
Risk	None

Table 3: Template for tactic and strategy description

agent (as discussed above). Tactic descriptions can also be compositional, so a tactic can make use of other tactics already specified. In this respect, STRATUM provides the designer with a *tactic or strategy template*, which is used to specify various tactics. A particular instantiation of this template is called a *tactic description* or *strategy description*. The tactic template, instantiated with the tactic described in section 3.4.2, is described in Table 3.

Each tactic description specifies the dialogical objective it is aimed at achieving. The *method* cell contains an informal description of the different steps the tactic involves in order to achieve its dialogical objective. The *capabilities and sub-tactics used* to execute these steps are specified in the following cell. Finally, the *rationality condition* specifies any constraints on the rational use of the tactic. For example, one may specify that it is only rational to make an offer to a counterpart if this offer is more preferred to the counterpart than any previously made offer. Finally, the designer can list the potential risks of enacting the strategy. This would enable the programmer of the strategy to take these risks into account and provide suitable solutions or precautions.

The process of designing strategies using tactics is similar to designing tactics using primitive capabilities. We follow the intuitive distinction between *strategies*, which govern an entire interaction or large parts of it, and *tactics*, which govern just a small number of utterances in an interaction. Hence, one tactic may support multiple or competing strategies. For example, asking a direct question may implement a strategy to gather information from another participant or it may implement a strategy to delay resolution of the negotiation, or both. The methodology leaves to the designer the decision about whether to refer to something as a "tactic" or a "strategy."[6] One might argue that this renders the

---

[6] In the latter case, the template above would be referred to as a *strategy template*.

distinction between tactics and strategies redundant. Nevertheless, this conceptual distinction may be useful from the strategy designer's point of view. Indeed, this distinction has proved useful, for example, in the work on natural language generation (McKeown, 1985).

3.5   Stage V: Test and Refine

The final stage of the methodology is to test the strategy in terms of some appropriate criteria or metrics. This testing may also lead to an iterative process of refinement, based on the new findings.

One way to perform testing is through *empirical* analysis. This would involve running simulations of different dialogues using a variety of strategies. Simulation parameters could be varied in a number of ways: among different strategies operating against a fixed type of counterpart, among different types of low level tactics within a particular generic strategy, in terms of the types of counterparts faced, the information available to agents, or the agents' time constraints. Such simulations have started to be used to analyse bargaining-based (Faratin, 2000), auction-based (He et al., 2003) and (to a lesser extent) argument-based strategies (Ramchurn et al., 2003).

Another way to test strategies is through theoretical analysis. In game-theory, this is normally done using equilibrium analysis, though this approach is usually aimed at designing the mechanism, not the strategy. For strategies that cannot be studied using traditional game-theoretic techniques, it is often hard to analyse strategies theoretically. Hybrid theoretical/empirical approaches have been used, where the outcomes under different strategies are first generated empirically, then studied analytically using game-theoretic concepts. This approach has been used, for example, by Fatima et al. (2004) to study bargaining in incomplete information settings. In economics, *behavioural* game theory (Camerer, 2003) uses a hybrid approach whereby simulation experiments are first designed and run. Then, regularities in repeated interactions are explored and studied analytically to identify *evolutionary* equilibrium conditions.

3.6   Summary of the Methodology

The STRATUM methodology is summarised in Figure 4. Stages I, II and III produce the OM, CM and EM respectively. Then Stage IV involves the simultaneous decomposition of objectives and composition of capabilities in order to produce tactics that eventually form strategies. Finally, the testing and refinement stage leads to either (i) adjusting the environment and counterpart model, or (ii) repeating Stage IV with different decompositions of objectives and compositions of capabilities.

It is worth noting that the clarity of the given multi-agent system specification influences the ease in which the OM, CM and EM can be captured by the negotiating agent designer. In some situations, the designer is given explicit specifications of the objectives, the agents' capabilities, and the environment in which agents operate, making the OM, CM and EM easy to capture. This is the case, for example, when the protocol is specified clearly in terms of declarative rules, or where the counterparts are fully
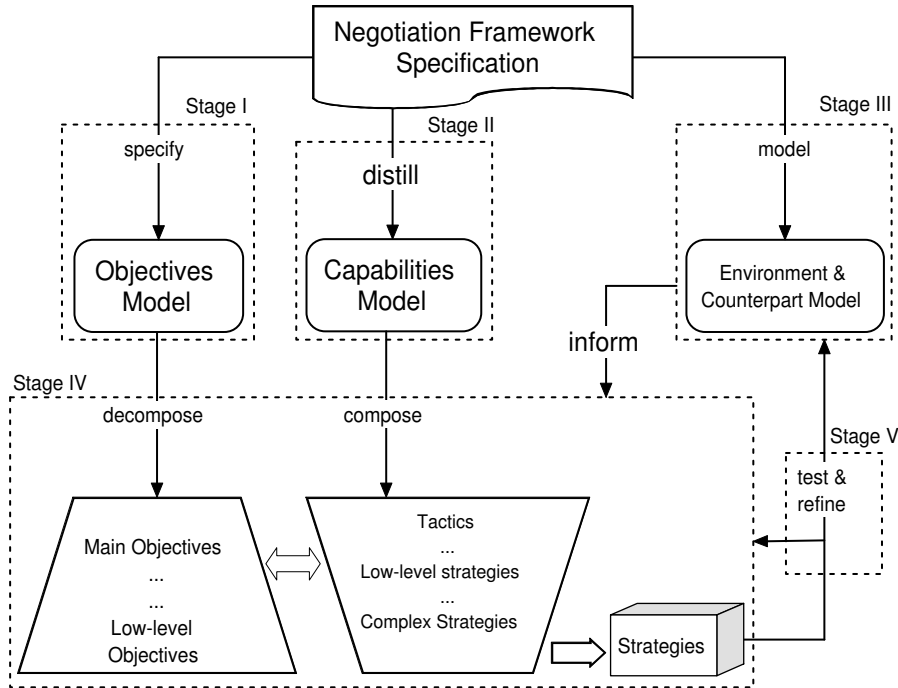
Figure 4: Abstract view of the methodology

predictable (e.g., in complete information settings). On the other hand, if the multi-agent system is not thoroughly specified, the strategy designer needs to make these implicit models explicit.

4 Applying the Methodology

In this section, we demonstrate, through two case studies, how the STRATUM methodology can be used to aid the design of tactics and strategies in particular negotiation frameworks. In the first case study, we explore an existing strategy used in the Trading Agent Competition (TAC, 2003) and demonstrate that STRATUM captures notions that the designers have included in the design of their strategy. In the second case study, we present another case study which addresses a different style of negotiation; namely interest-based negotiation (IBN) between two agents (Rahwan, 2004).

We selected these two domains for our case studies because they exemplify significantly different approaches to automated negotiation. The TAC domain involves multiple concurrent negotiations using a variety of auction-based protocols. Here, the focus is on modelling the general dynamics of the environment and reacting accordingly. The IBN domain, on the other hand, involves a richer communication language based on computational models of argumentation. Here, the focus is on modelling the negotiation counterpart and planning an argumentation strategy to persuade the opponent of changing its beliefs and goals.

It is worth noting that we do not undertake a thorough and complete application of the STRATUM

methodology in the case studies below. Instead, our aim is to give the reader a feel for how the different stages of the methodology can be applied in particular, relatively rich, negotiation scenarios.

4.1 Case Study 1: Trading Agent Competition

We now *deconstruct* an existing strategy used by an agent participating in the Trading Agent Competition (TAC, 2003). Through this "rational deconstruction," our aim is to demonstrate that STRATUM captures notions that the designers of this strategy have reasoned about as they designed their strategy. Our aim is that this deconstructive exercise will help support our claims about the feasibility of the methodology and its consistency with current actual practice. Specifically, the analysis is based on SouthamptonTAC (He and Jennings, 2003), a trading agent developed at the University of Southampton, and one of the most successful agents in TAC2002.

We start with a brief description of TAC games.[7] In a TAC game, there are eight software agents that compete against each other in a variety of auctions to assemble travel packages for 64 customers (8 customers each). A package consists of (i) a round trip flight during a 5-day period between TACtown and Tampa; and (ii) a stay at a particular hotel for every night between their arrival and departure. Customer satisfaction is measured in terms of *utility*, and each agent attempts to maximise its customers' utility, as measured against customer preference data that is randomly generated at the start of the game. Agents can also bid for the *optional* entertainment package, which can provide additional utility. There are different types of entertainment options. An individual game lasts 12 minutes and involves 28 auctions. Each component in a package is traded in a different type of auction:

- *Flights* are sold in single seller auctions, for which the ask price is updated randomly every 24 to 32 seconds.

- *Hotels* are traded in 16th price multi-unit English auctions. There are eight hotel auctions that close in random order at the end of every minute after the 4th. When a hotel auction clears, it allocates 16 rooms to agents that bid the 16 highest prices.

- *Entertainment* tickets are randomly provided to agents, 12 tickets each, at the beginning of the game. Then agents can trade their tickets in a continuous double auction (CDA), where agents can buy and sell at any time before the game closes. Tickets are only useful to a customer if they are for different events, on different dates, and for nights when the customer is in town.

Designing bidding strategies for TAC auctions has proved to be a challenging problem. There are many interdependencies between different kinds of auctions (e.g. flights will be useless if the hotel rooms are not available) and within the same auction (e.g. having an extra ticket for the same entertainment is useless). This means that agents have to solve combinatorial optimisation problems in a very short time

---

[7]For the interested reader, Greenwald (2003) provides a comprehensive description.

(Eriksson and Janson, 2002). Moreover, there is an inherent uncertainty in the domain (e.g. since flight prices change randomly) and there is no way to predict precisely how other agents will behave.

Given this background, let us deconstruct SouthamptonTAC-02 based on STRATUM. **Stage I** (specifying objectives) is straightforward, since the objectives are explicitly stated in the form of the utility formula that takes into account complex customer preferences over flight, hotel and entertainment combinations (see (TAC, 2003) for details). Each agent attempts to maximise its overall utility based on the given customers' preferences.

Stage II (specifying capabilities) is again relatively simple. This is because the protocol allows relatively few dialogical capabilities. Agents have the ability C1 of making proposals in the form of bids according to the different auction protocol rules, the ability C2 to accept proposals in the entertainment CDAs, and the ability C5 to seek information from counterparts *indirectly* by observing their bidding behaviour. Finally, agents have the ability C11 (do nothing) by simply not making any bids. Agents are not allowed to leave the game before it is over, nor enter the game after it starts. Moreover, agents have no external non-dialogical abilities that may influence their dialogical abilities.

The authors of SouthamptonTAC-02 invested a significant effort into **Stage III** (constructing the Environment Model). In the authors own words:

> "Our post hoc analysis of the TAC2001 competition shows that an agent's performance depends heavily on the risk attitude of its opponents." (He and Jennings, 2003, page 221)

This marks an explicit recognition by the authors of the importance of modelling the negotiation environment (or more precisely, peer agents in the environment). To this end, the authors identify three types of environments, based on the prices of hotels: (i) *competitive environments* where the prices of the hotels are very high, (ii) *noncompetitive environments* where agents can get cheap hotel rooms, and (iii) *semicompetitive environments* which are in between. Environment type recognition is performed by monitoring the hotel prices during the game. Recognition is based on *fuzzy pattern recognition* techniques, which classify the environment based on the degrees of membership of the asking price in the fuzzy sets representing the three environment types mentioned above.

Note that the authors did not use the prices of entertainment tickets or flight tickets as a measure of competition. This is because they have observed that flight prices and entertainment tickets prices are similar whether the game is competitive or non-competitive. It is the hotel closing prices that identify the nature of the opponents.[8]

Let us now move to **Stage IV** (constructing tactics and strategies). Given the limited number of dialogical abilities allowed, tactics in TAC are mainly characterised by the following two families: *seeking to delay* and *seeking fast termination*. In other words, tactics are mainly time-dependent. The precise way in which agents delay or speed up their buying and selling, and the reasons for doing so (e.g. based on price prediction), are what constitutes TAC strategies.

---

[8]Based on personal communication with the designers of SouthamptonTAC-02 (He, 2004).

The authors begin with the main objective in mind: to maximise the utility. Then, they divide this objective into the following sub-objectives: (i) predicting the hotel closing prices; (ii) allocating flights, hotels and entertainment tickets obtained to the eight customers so that utility is maximised; (iii) deciding when to bid in the flights auctions; (iv) deciding what entertainment tickets to buy and sell in the continuous double auctions; and (v) deciding what hotel auction to bid in and at what price (He, 2004).

Note that objectives (i) and (ii) above are not dialogical objectives since they involve internal calculations by the agent rather than interaction with other agents. The first objective is achieved using fuzzy rules, while the second is achieved using an integer and linear programming solver. Among the three remaining objectives, our interest in the remainder of this section is in the strategy used to achieve objective (iii) for deciding when to bid for flight tickets.

Based on observations during the earlier TAC-01 competition, the authors of SouthamptonTAC-02 recognised the need for adapting to the different types of environments described above. The authors characterise a *risk-averse* (RA) tactic as one where the agent buys a small number of flight tickets at the beginning of the game and bids for hotels according to the situation as the game progresses. A *risk-seeking* (RS) tactic, on the other hand, is one where the agent buys a large number of flight tickets at the beginning of the game, and therefore does not change its customers' travel plans often during the game. The authors state the following observation about the risk-averse tactic, based on their experience in TAC2001:

> "a *risk-averse* agent ... is highly flexible and copes well when there is a significant degree of competition and the hotel prices are high ... In this way, it avoids buying extra hotels which cost extra money. Also, the agent can receive optimal utility by not shortening the stay of its customers." (He and Jennings, 2003, pages 221, 226)

The authors also make the following observation about the risk-seeking strategy:

> "a *risk-seeking* agent ... does well in environments in which hotels are cheap. For example, when a hotel price goes up sharply, a risk-averse agent would stop bidding on that hotel (changing the stay to a counterpart hotel or reducing the trip period). In contrast, a risk-seeking agent will insist on bidding for that hotel, although the price is very high. In so doing, it hopes that the price will eventually stabilise (hence the risk) ... (It) is highly effective in noncompetitive environments ... because there is little competition in hotel bidding and the agent can always obtain what it wants." (He and Jennings, 2003, page 222, 226)

It is possible to characterise the above observations, which guided the design of the SouthamptonTAC-02 strategy, using STRATUM tactic templates. The risk-averse and risk-seeking tactics are encoded in Tables 4 and 5 respectively.

Stage IV of strategy design does not stop here. The authors make the following additional observation:

Tactic Name	Risk-Averse Tactic (RA)
Dialogical Objective	- Avoid buying extra expensive hotel bookings - Optimise utility by getting the lengths of customers' visits right
Method	- Buy small number of flight tickets at the beginning; - Bid on hotels if prices are not very high; Otherwise, switch to the other hotel type or change the journey;
Capabilities & Sub-Tactics Used	- Ability C1 (making proposals/bids) - ability C5 (seek information) by observing other agents' bidding behaviour - ability C11 (do nothing)
Rationality Condition	Environment is competitive
Risk	If the environment is noncompetitive, the agent misses out on opportunities to optimise length of trips

Table 4: Risk-averse tactic specification

Tactic Name	Risk-Seeking Tactic (RS)
Dialogical Objective	Optimise holiday allocation, and take advantage of lack of competition in order to get the hotels it wants
Method	- Buy large number of flight tickets at the start - Purchase suitable hotels later, without changing flight bookings
Capabilities & Sub-Tactics Used	- Ability C1 (making proposals/bids) - ability C5 (seek information) by observing other agents' bidding behaviour - ability C11 (do nothing)
Rationality Condition	- Environment is noncompetitive - If hotel prices rise, they eventually stabilise
Risk	- If environment gets competitive, hotel prices rise significantly; as a result, one must either pay high hotel prices, or change travel plans and shorten stay, hence wasting travel tickets already bought

Table 5: Risk-seeking tactic specification

Tactic Name	SouthamptonTAC-02 Adaptive Strategy
Dialogical Objective	Maximise utility by adapting to changes in the environment
Method	- If environment is competitive, use the risk-averse tactic - If environment is noncompetitive, use risk-seeking tactic - If environment is semicompetitive, use medium-risk tactic
Capabilities & Sub-Tactics Used	- Risk-Averse Tactic - Risk-Seeking Tactic - Medium-Risk Tactic
Rationality Condition	–
Risk	–

Table 6: SouthamptonTAC-02 adaptive strategy

"After our experiences in TAC-01, we came to believe that there is no single best strategy that can deal with all the different types of TAC environment. For example, a risk-seeking agent ... is highly effective in noncompetitive environments. This is because there is little competition in hotel bidding and the agent can always obtain what it wants. On the other hand, delaying buying flights and shortening the stay of customers[9] works well in competitive games. For this reason, SouthamptonTAC dynamically varies its bidding strategy according to its assessment of the environment type." (He and Jennings, 2003, page 226)

Based on this observation, SouthamptonTAC-02 uses a *composed* strategy which makes use of the risk-seeking and risk-averse tactics, as well as a *medium-risk tactic* in semicompetitive environments.[10] This can be characterised in the STRATUM strategy template shown in Table 6.

Finally, for **Stage V** (testing and refinement), SouthamptonTAC-02 strategy testing has been mainly based on simulation. The authors set up a number of controlled experiments in their lab. In particular, they simulated games in which different combinations of SouthamptonTAC-02, risk-averse and risk-seeking agents played (among 8 participants). They showed that SouthamptonTAC-02 does best in competitive games, where the number of RS agents is big. The agent also does well in noncompetitive environments, where there are many RA-agents. It turns out that the worst situation for SouthamptonTAC-02 is when all players are like itself, i.e. they are all adaptive. This is due to the fact that all agents switch their strategies (i.e. adapt) at the same time, causing competition to be instantaneously shifted, or for prices to fluctuate constantly.

Further empirical results were obtained in the actual TAC2002 competition. TAC2002 results showed that SouthamptonTAC-02 was ranked second, with a difference of 0.8% from the top scoring agents.

---

[9]I.e. a risk-averse behaviour

[10]In this tactic, the agent buys most of the flights earlier and will only change travel plans if a significant improvement can be obtained.

However, since each game involves a wide variety of agents, it is difficult to draw strong comparative results, such as whether the adaptive strategy made a real difference.

In terms of *strategy refinement* that took place *before* the actual competition, the authors denote that it was due to empirical testing through controlled experiments that they discovered the need for modelling the environment (He, 2004). This is what led them to using the fuzzy-logic-based method for modelling the environment's competitiveness. The authors also indicated that their future work includes improving the fuzzy modelling and pattern matching techniques (He and Jennings, 2004).

One possible future refinement could be to attempt to deal with the problem that arises when multiple identical adaptive agents populate the game. In such cases, adding some randomness to the timing of agents' adaptation (e.g. by allowing an agent to switch its strategy before others do) might enable an agent to overcome the problem mentioned above. Another refinement suggestion is to include other factors when modelling the environment. For example, instead of considering hotel prices as the only measure of environment competitiveness, the strategy designers may take account of competitors' bidding behaviour in entertainment and travel auctions.

4.2 Case Study 2: Interest-based Negotiation

Interest-based Negotiation (IBN) (Rahwan, 2004, Chapter 3) is a type of argumentation-based negotiation frameworks (Rahwan et al., 2003). We begin by giving a brief overview of IBN before we discuss how STRATUM may be applied to it.

The idea behind IBN is that in addition to exchanging proposals, acceptance and rejection messages, agents can also exchange information about their underlying goal structures and use this information in order to discover better deals. The IBN framework assumes agents are built using a deliberative architecture where they have *beliefs* about the environment and *capabilities* that define what non-dialogical actions they can perform. Agents have *desire generation rules* of the form $C \Rightarrow \mathsf{d}(\phi)$, which means that if condition C is satisfied, then the agent will *desire* to achieve ϕ. Using desire generation rules, an agent can generate *explanatory arguments* (Amgoud and Kaci, 2005) by which they conclude which desires to adopt. *Planning rules*, on the other hand, are expressions of the form $\phi_1 \wedge \cdots \wedge \phi_n \rightarrowtail \phi$ and mean that if $\phi_1, \ldots, \phi_n$ are achieved, then ϕ is achieved. Using planning rules, an agent can generate *plans* (or *instrumental arguments*) for achieving different desires. These plans have a hierarchical (tree) structure in which the root node is a desire, all other nodes are *goals* that are *instrumental* towards achieving the desire, and leaf nodes represent the *resources* required to execute the plan (i.e. required non-dialogical capabilities). The relationship between beliefs, desires, goals and resources is depicted in Figure 5.

Agents calculate the *utilities* of different plans based on the plans' *costs* and the *values* of the desires they achieve, resulting in a preference ordering over different plans. As agents receive new information from the environment or other agents, their existing arguments may be *defeated*, resulting in change in their preference. An argumentation-based semantics is given that specifies how preference changes
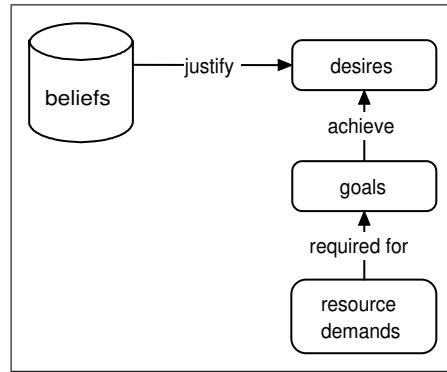
Figure 5: Relationships between attitudes

occur.[11]

The IBN protocol provides a way for agents to exchange information about their goal structures, and to exchange arguments that can alter these goal structures. Locutions PROPOSE(.), ACCEPT(.) and REJECT(.) allow agents to propose, accept and reject deals, respectively. The locution ASSERT(.) allows an agent to make assertions about its own beliefs, goals, sub-goals or planning rules. Agents can ask one another for information through the locutions QUESTION(.), which allows an agent to ask another to state whether it agrees with a particular expression, and CHALLENGE(.), which allows an agent to ask another for a *justification* of a particular asserted belief or desire. The locution RETRACT(.) enables agents to retract previously asserted statements and proposed deals.

Note that locution CHALLENGE(.) allows agents to either ask for arguments (i.e. tentative proofs) that justify a particular "belief" or ask for explanatory arguments that justify a particular "desire." This cannot be used to ask for arguments supporting *goals* that are part of instrumental arguments. This is because a goal is adopted both because it is *instrumental* towards a higher-level goal or desire, and because it is *achievable* through some lower-level goals or capabilities. For this purpose, there are two additional locutions that enable agents to ask for arguments supporting goals. Location REQ-PURPOSE(.) allows an agent to ask another for the higher-level purpose of a particular request (e.g. "why do you need to travel?"). A similar locution REQ-ACHIEVE(.) allows an agent to ask another for the sub-plan that is meant to achieve the goal in question (e.g. "how do you intend to book your ticket?").

The framework also has additional special predicates to use within the locutions. The predicate prule(.) is used to exchange planning rules among agents. Another predicate $instr(X, Y)$ is used by one agent to indicate that goals X are adopted because they are *instrumental* towards achieving higher-level goals Y. Predicates des(.), bel(.) and int(.) denote the desire, belief and intention modalities, respectively.

Finally, the framework uses so-called *commitment stores* (Walton and Krabbe, 1995). Each agent i has its own commitment store $\mathcal{CS}(i)$, which stores commitments that follow from different utterances

---

[11]For brevity, we do not present the full specification of the argumentation system used. The interested reader may refer to (Rahwan, 2004).
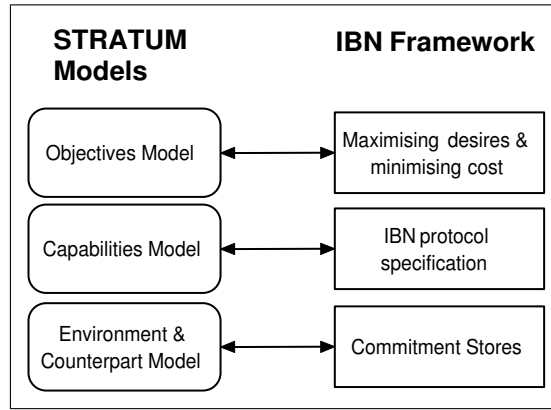
Figure 6: Methodology models mapped in IBN

made by that agent. For example, when an agent makes an assertion, this assertion is inserted into its commitment store. This enables other agents to later question or challenge these assertions. Also, when a proposal is made, a conditional intention is inserted into the proposer's commitment store. The conditional intention states that if the counterpart agrees on its part of the deal, then the proposer is committed to intending its own part.

Stage I of the STRATUM methodology requires identifying the agent's objectives in the negotiation. Objectives are specified explicitly in the IBN framework, making the OM easy to document. Each agent attempts to achieve a set of desires and tries to reach a deal that maximises the utility it receives (calculated as the difference between the worth of desires achieved and the cost of the actions needed).[12]

Stage II of the methodology requires defining the agent's capabilities, which constitute the CM. These are also explicitly specified already. In terms of the typology in table 1, the IBN protocol presented in (Rahwan, 2004) enables, in some form or another, all capabilities except C7 (exerting pressure on a counterpart). Agents can propose, accept and reject deals; they can also make assertions, retract commitments, ask each other for information about each others' beliefs, or ask questions about how they may achieve certain goals, or what a particular goal is useful for. They can also do nothing by uttering the PASS locution. Note, however, that the exact type of capability is somewhat restricted. For example, while an agent can request information about the counterpart's higher-level and lower-level goals (through the REQ-PURPOSE(.) and REQ-ACHIEVE(.) locutions) and about the counterpart's beliefs (using the QUESTION(.) and CHALLENGE(.) locutions), the agent is not able to ask the counterpart about the reason for believing a certain planning rule.

Moving to **Stage III** (constructing the Environment Model EM), we first know that both agents use the same underlying reasoning mechanisms. This is an important assumption, since if the counterpart does not have an explicit representation of underlying goals, for example, then it does not make sense to ask that counterpart about the purpose of a particular request. In IBN, an explicit model of the counterpart

---

[12]Note that this uses a different, and simpler formula, to the one used in TAC.

is available at any time through its commitment stores. One may also attempt to *guess* other information about the counterpart by inferring its beliefs, goals etc. from its utterances. This would benefit from the assumption that all agents have symmetric reasoning abilities. But for the purpose of this discussion, commitment stores are sufficient. Figure 6 summarises the mapping between the IBN framework and the first three models of the STRATUM methodology.

Stage IV involves defining tactics and strategies from capabilities, in light of some decomposition of objectives. We now give an example of the kinds of analysis involved in Stage IV. Suppose we attempt to design a strategy for our seller agent A_1 to negotiate with another buyer agent A_2. As strategy designers, we would like to design a particular strategy that contributes to A_1's objective, and we want this strategy to be suitable for a situation where A_2 has already proposed some deal D to our agent, and that D is not acceptable to A_1. One way to achieve A_1's negotiation objective is to get A_2 to accept another deal D' which is preferred by A_1. The most trivial strategy to achieve this is to offer D' to A_2. This is clarified in the following typical bargaining dialogue:

A2: $\mathsf{PROPOSE}(A_2, A_1, D)$
A1: $\mathsf{REJECT}(A_1, A_2, D)$
A2: $\mathsf{PASS}(A_2)$
A1: $\mathsf{PROPOSE}(A_1, A_2, D')$

If the above attempt fails, the only potential solution is to propose other alternative deals D'', D''' etc. However, IBN enables alternative ways to reach a solution, by providing agents with a wider set of dialogical abilities. Reasoning about how these dialogical capabilities may influence the outcome requires an understanding of the effect of these capabilities on preferences. Knowing that in IBN, both agents are assumed to have identical reasoning models, we can make the following observations about the counterpart:

- A_2 might have rejected D' because it has some goal g and it believes that D' does not achieve g;

- If A_2 is persuaded that D' does actually achieve g, then it might accept D' (subject to other issues, such as the costs involved);

Hence, based on the above understanding of the counterpart, we can construct a strategy that attempts to persuade A_2 that D' does indeed achieve its goal(s). But first, we need to explore how the IBN agent capabilities can be composed to achieve such persuasion. Given that A_1 has the ability C5 to seek information about the purpose of a resource requested by A_2, and to present new information about planning rules, we can construct the following pattern to persuade A_2 to accept D':

- A_1 finds out about a goal g that A_2 wants to achieve using resource r, which is part of deal D that A_2 requested;

- A_1 then argues that D' can also achieve goal g;

Strategy Name	S1
Dialogical Objective	Cause counterpart to intend a deal that is more preferable to me
Method	- Find out what deal the counterpart currently wants, call it D - Find out what goal D is intended to achieve; - Assert that D' also achieves that goal; - Offer deal D'
Capabilities & Sub-Tactics Used	- Capability C5 (seek information) - Capability C4 (present information proactively) - Capability C1 (make a proposal)
Rationality Condition	- Deal D' must be acceptable to me, and more preferred to me than deal D
Risk	–

Table 7: Description of an example IBN strategy

– A_1 offers D' to A_2;

This strategy can be described in STRATUM as shown in table 7. The following dialogue sequence demonstrates how this strategy can be encoded in the IBN protocol:

A2: $\mathsf{PROPOSE}(A_2, A_1, D)$

   this leads to inserting $\mathsf{int}(r)$ for each $r \in Resources(D)$ to A_2's commitment store $\mathcal{CS}(A_2)$, where $Resources(D)$ denotes the resource required in deal D

A1: $\mathsf{REJECT}(A_1, A_2, D)$

A2: $\mathsf{PASS}(A_2)$

A1: $\mathsf{REQ\text{-}PURPOSE}(A_1, A_2, r)$

   where $r \in D$

A2: $\mathsf{ASSERT}(A_2, A_1, \mathsf{instr}(r, g))$

   this leads to $\mathsf{int}(g)$ being inserted to $\mathcal{CS}(A_2)$

A1: $\mathsf{ASSERT}(A_1, A_2, \mathsf{prule}(r_1 \wedge \cdots \wedge r_n \rightarrowtail g))$

A2: $\mathsf{PASS}(A_2)$

A1: $\mathsf{PROPOSE}(A_1, A_2, D')$

   where $r_1, \ldots, r_n \in D'$

If the strategy works, the above dialogue will be followed by an acceptance from agent A_2 using the utterance $\mathsf{ACCEPT}(A_2, A_1, D')$. The following is a natural language description of a dialogue that uses the above strategy.

A1:   *Why do you need the car?*

A2:   *To travel to Sydney.*

Tactic Name	T1
Dialogical Objective	Cause counterpart to abandon some intended deal D
Method	- Ask the counterpart whether it intends some goal g'; - Argue that D has the undesirable side-effect of precluding g'
Capabilities & Sub-Tactics Used	- Ability to provide information about consequences
Rationality Condition	–
Risk	–

Table 8: Tactic description of an IBN tactic

A1: *You can also go to Sydney by flying there.*

A2: *But this would be even more expensive.*

A1: *We're in the low-season, so I can book you a flight for only $250.*[13]

A2: *That's great. Go ahead and book it please.*

In the above example, agent A_1 simply presents information about an alternative way of achieving the goal of going to Sydney. Whether A_2 accepts this alternative might depend on other issues, such as whether A_2 has petrol vouchers, or whether A_2 wishes to visit a friend who lives on the way to Sydney. Therefore, a more aggressive strategy would be to try to make the driving alternative itself less appealing. As a result, this might make the flying option more preferable, hence increasing the likelihood of its acceptance. This new new approach can be described by the following pattern:

- A_1 finds out about a goal g that A_2 wants to achieve using resource r, which is part of deal D that A_2 requested;

- A_1 then argues that D' can also achieve goal g;

- A_1 persuades A_2 that its previous deal D does not achieve goal g;

- A_1 offers D' to A_2;

Note that this strategy is similar to strategy S1, except that it requires A_1 to persuade A_2 to abandon D. So before we specify the new strategy, we need to specify a tactic for achieving such persuasion. One way to do so is to show a "side effect" of D that was not known to A_1 before. This tactic is described in table 8, and is self explanatory. We can now describe the new strategy by referring to tactic T1. This strategy is described in table 9.

The following is an illustration of how strategy S2 can be encoded in the IBN protocol.

---

[13]Of course, this dialogue assumes that A_1 prefers to sell a ticket for $250 than hire out a car for $400.

Strategy Name	S2
Dialogical Objective	Cause the counterpart to intend a deal that is more preferable to me
Method	- Find out what deal the counterpart currently wants, call it D; - Find out what goal D is intended to achieve, call this goal g; - Assert that D' also achieves that goal; - Cause counterpart to no-longer intend D; - Offer deal D'
Capabilities & Sub-Tactics Used	- Capability C5 (seek information) - Capability C4 (present information proactively) - Tactic T1 to cause counterpart to abandon D - Capability C1 (make a proposal)
Rationality Condition	- Deal D' must be acceptable to me, and more preferred to me than deal D
Risk	- After abandoning deal D, the counterpart may still not accept deal D', for some other reason. As a result, if our agent prefers accepting D to nothing, and it is not possible to persuade the counterpart to intend D again, then our agent is worse off

Table 9: Description of an example IBN strategy

A2: PROPOSE(A_2, A_1, D)

   this leads to inserting $\mathsf{int}(r)$ for each $r \in Resources(D)$ to A_2's commitment store $\mathcal{CS}(A_2)$

A1: REJECT(A_1, A_2, D)

A2: PASS(A_2)

A1: REQ-PURPOSE(A_1, A_2, r)

   where $r \in D$

A2: ASSERT($A_2, A_1, \mathsf{instr}(r, g)$)

   this leads to $\mathsf{int}(g)$ being inserted to $\mathcal{CS}(A_2)$

A1: QUESTION($A_1, A_2, \mathsf{int}(g')$)

A2: ASSERT($A_2, A_1, \mathsf{int}(g')$)

A1: ASSERT($A_1, A_2, \mathsf{prule}(r_1 \wedge \cdots \wedge r_n \rightarrowtail \neg g')$)

   where $r_1 \wedge \cdots \wedge r_n \subseteq Resources(D)$

A2: PASS(A_2)

A1: PROPOSE(A_1, A_2, D')

   where $r_1, \ldots, r_n \subsetneq Resources(D')$

The following follow-up natural language dialogue illustrates the usage of the strategy:

A2: *I still prefer to drive to Sydney.*

A1: *Are you collecting frequent flyer points?*

A2:   *Yes!*

A1:   *Because you are dropping the car in another state, you will not be able to get frequent flyer points by hiring this car.*

A2:   *Oh! I thought I would. In that case, I prefer to fly and get the points.*

The risk associated with strategy S2, as shown in table 9, is clarified in the following variant of the above dialogue:

A2:   *I still prefer to drive to Sydney.*

A1:   *Are you collecting frequent flyer points?*

A2:   *Yes!*

A1:   *Because you are dropping the car in another state, you will not be able to get frequent flyer points by hiring this car.*

A2:   *Oh! I thought I would. In that case, I don't want to hire a car, and I'd rather stay in Melbourne for my holiday; I hate flying.*

Finally, testing and refinement of the strategies is performed in **State V**. This stage on its own may require a separate study, because of the richness of the protocol and the complexity of possible dialogue sequences. However, we shall briefly discuss some possible directions.

One option is to perform empirical testing through simulation. In argument-based negotiation, such types of studies have only been used for very simplistic dialogues (Jung et al., 2001; Karunatillake and Jennings, 2004; Ramchurn et al., 2003), where variations in parameters are relatively easy to enumerate (e.g. by varying the strength of threats and promises based on trust). The lack of extensive research in this area is largely due to the complexity of the protocol, and the fact that no generic formal theory of argument-based interaction protocols exists. The STRATUM methodology has the potential to enable designers of multi-agent systems to follow a more systematic approach when enumerating variations of strategy within richer protocols.

The other option for analysing IBN strategies is to perform theoretical analysis. This approach has been used to study simple strategies in persuasion dialogues (Amgoud and Maudet, 2002), as well as in inquiry and information seeking dialogues (Parsons et al., 2002), but not in negotiation dialogues as yet. Torroni (2002) studies whether certain strategies lead to dialogue termination. The formal analysis is facilitated by the fact that the system is specified using Abductive Logic Programs (Sadri et al., 2002). This enabled the analysis of dialogues in relation to the underlying well-understood proof-theory. This type of formal analysis may not be as simple for the IBN framework, since many parts of the framework are described in a more or less "semantic" fashion (e.g. the generation of candidate intentions).

4.3 Discussion

The long-term objective of the research reported in this paper is to bridge the gap between the theory and practice of automated negotiation. To this end, it is important to identify the domain-independent aspects of negotiating agent design in order to come up with general principles that can guide software engineers in building such agents.

The case studies above provide a hands-on feel for how the methodology can be applied. They also demonstrate how STRATUM can provide guidance to the process of strategy design in reasonably complex, and significantly different negotiation environments. The first case study shows that our methodology is consistent with current practice, since we were able to *rationalise* an existing TAC strategy using STRATUM. The second case study, on the other hand, shows that the same primitives and processes can be applied to design strategies for a very different type of negotiation environment, namely one that involves an expressive argumentation-based protocol.

The applicability of STRATUM in such significantly different domains demonstrates that the methodology's primitives and processes are sufficiently general to be applicable in a wide range of settings. In this context, we note that this wide applicability of STRATUM was facilitated by our deliberate choice to keep the methodology informal, rather than using a specific formal language.

We also believe the methodology has the potential to provide a common framework for comparison of strategies across negotiation frameworks, hence guiding designers in selecting appropriate strategies based on distilled domain characteristics. For example, the adaptive strategy described in case study 1 above may be applicable in other domains where an agent is able to observe other agents' bidding behaviour. Similarly, whenever the protocol enables agents to seek information about other agents' goals, strategy S1 (See Table 7) may be applied in order to accelerate the process of finding a mutually acceptable deal. Hence, the paper contributes to a vision discussed in an earlier paper (Jennings et al., 2001):

> "...there are also a number of broader issues, which, to date, have received comparatively little attention. These include ...the development of a best practice repository for negotiation techniques. That is, a coherent resource that describes which negotiation techniques are best suited to a given type of problem or domain (much like the way design patterns function in object-oriented analysis and design)." (Jennings et al., 2001, page 212)

5 Conclusion and Future Work

We have presented the STRATUM methodology for guiding the design of agent negotiation strategies. STRATUM is the first attempt to structure the process of strategy construction in non-game-theoretic domains (i.e. in domains where the nature of the protocol and the limited information available make it impossible to prescribe optimal strategies through mechanism design techniques (Dash et al., 2003)). We demonstrated how the methodology can be used through two case studies. Moreover, we believe there is

nothing in our model that is specific to one type of negotiation framework and so we believe it is generic. However, additional studies are needed to better understand and confirm the scope of applicability of the approach.

The hierarchical construction of strategies becomes particularly useful for designers of strategies for rich negotiation dialogues, where agents have a large number of possible utterances (and hence, dialogical abilities) that they can use. This is because STRATUM enables designers to reason about the objectives of a negotiation strategy as a planning problem. This view of dialogue is consistent with the plan-based theory of speech acts proposed by Cohen and Perrault (1979), which suggests that a "planning" view is an adequate way to capture the way people reason about their utterances.

It is worth noting that it may be possible to exploit the hierarchical structure of objectives and strategies in order enable agents to compose strategies at run-time. In fact, this is exactly what SouthamptonTAC-02 does, since it varies its underlying tactic based on observations of the environment. In more complex dialogues, this would require the dependencies between objectives and tactics on different levels to be represented formally in a way that can be processed by a computer program. Then the reasoning required may be made possible using a hierarchical task planner (Erol et al., 1994). This would enable agents to modify their strategies, or even their objectives, during negotiation. For example, an agent might abandon an objective or a tactic if the agent perceives that this objective or tactic is not currently achievable or is counterproductive because the negotiation counterpart is resistant to it. As with any other intentions, the defeasibility of objectives in a computational agent requires some formal mechanism for intention-reconsideration (Schut and Wooldridge, 2001).

Acknowledgements

References

Amgoud, L. and S. Kaci (2005). On the generation of bipolar goals in argumentation-based negotiation. In I. Rahwan, P. Moraitis, and C. Reed (Eds.), *Argumentation in Multi-Agent Systems: (Proceedings*

of the First International Workshop (ArgMAS'04): Expanded and Invited Contributions), Volume 3366 of *Lecture Notes in Computer Science*. Berlin, Germany: Springer Verlag.

Amgoud, L. and N. Maudet (2002). Strategical considerations for argumentative agents (preliminary report). In S. Benferhat and E. Giunchiglia (Eds.), *Proceedings of the 9th International Workshop on Non-Monotonic Reasoning (NMR 2002): Special session on Argument, Dialogue and Decision*, pp. 399–407.

Ashri, R., I. Rahwan, and M. Luck (2003). Architectures for negotiating agents. In V. Marik, J. Müller, and M. Pechoucek (Eds.), *Multi-Agent Systems and Applications III: Proceedings of the 3rd International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2003)*, Volume 2691 of *Lecture Notes in Artificial Intelligence*, pp. 136–146. Berlin, Germany: Springer Verlag.

Bartolini, C., C. Preist, and N. R. Jennings (2002). Architecting for reuse: A software framework for automated negotiation. In F. Giunchiglia, J. Odell, and G. Weiss (Eds.), *Proceedings of the 3rd International Workshop on Agent-Oriented Software Engineering (AOSE-2002)*, Berlin, Germany. Springer Verlag.

Bauer, B., J. Müller, and J. Odell (2001). Agent UML: A formalism for specifying multiagent interaction. In P. Ciancarini and M. Wooldridge (Eds.), *Agent-Oriented Software Engineering*, Volume 1957 of *Lecture Notes in Computer Science*, pp. 91–103. Berlin, Germany: Springer Verlag.

Billington, D. (1993). Defeasible logic is stable. *Journal of logic and computation 3*, 370–400.

Binmore, K. and N. Vulkan (1997, April). Applying game theory to automated negotiation. In *Proceedings of the DIMACS workshop on economics, game theory and the Internet*, New Brunswick NJ, USA. Rutgers University.

Boella, G. and L. van der Torre (2004). Groups as agents with mental attitudes. In N. R. Jennings, C. Sierra, L. Sonenberg, and M. Tambe (Eds.), *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Washington DC, USA, pp. 964?971. IEEE Computer Society.

Bresciani, P., A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos (2004). Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems 8*(3), 203–236.

Camerer, C. F. (2003). *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton NJ, USA: Princeton University Press.

Cohen, P. R. and C. R. Perrault (1979). Elements of a plan-based theory of speech acts. *Cognitive Science 3*(1), 177–212.

Dash, R. K., D. C. Parkes, and N. R. Jennings (2003). Computational mechanism design: A call to arms. *IEEE Intelligent Systems 18*(6), 40–47.

Dignum, V. (2004). *A model for organizational interaction: based on agents, founded in logic.* Ph. D. thesis, Institute of Information and Computing Sciences, Utrecht University, The Netherlands.

d'Inverno, M. and M. Luck (2001). *Understanding Agent Systems.* Springer Verlag.

Dumas, M., G. Governatory, A. H. M. ter Hofstede, and P. Oaks (2002). A formal approach to negotiating agent development. *Electronic commerce research and applications 1*(2).

Dunne, P. E. (2003). Prevarication in dispute protocols. In G. Sartor (Ed.), *Proceedings of the 9th Internatonal Conference on AI and Law (ICAIL)*, New York, NY, USA. ACM Press.

Eriksson, J. and S. Janson (2002, October). The trading agent competition – TAC 2002. *ERCIM News* (51), 57–58.

Erol, K., J. Hendler, and D. Nau (1994). Semantics for hierarchical task network planning. Technical Report CS-TR-3239, UMIACS-TR-94-31, Department of Computer Science, University of Maryland.

Faratin, P. (2000). *Automated Service Negotiation Between Autonomous Computational Agents.* Ph. D. thesis, University of London, Queen Mary and Westfield College, Department of Electronic Engineering.

Fatima, S., M. J. Wooldridge, and N. R. Jennings (2001). Optimal negotiation strategies for agents with incomplete information. In J.-J. Meyer and M. Tambe (Eds.), *Intelligent Agent series VIII: Proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages (ATAL 2001)*, Volume 2333 of *Lecture Notes in Computer Science*, pp. 53–68. Berlin, Germany: Springer Verlag.

Fatima, S., M. J. Wooldridge, and N. R. Jennings (2002). Multi-issue negotiation under time constraints. In C. Castelfranchi and L. Johnson (Eds.), *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, New York, USA, pp. 143–150. ACM Press.

Fatima, S., M. J. Wooldridge, and N. R. Jennings (2004). Bargaining with incomplete information. *Annals of Mathematics and Artificial Intelligence.* (to appear).

FIPA (2001, 10 August). Communicative Act Library Specification. Technical Report XC00037H, Foundation for Intelligent Physical Agents.

Fisher, R., W. Ury, and B. Patton (1991). *Getting to Yes: Negotiating Agreement Without Giving In* (second ed.). New York, USA: Penguin Books.

Fritschi, C. and K. Dorer (2002). Agent-oriented software engineering for successful TAC participation. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 45–46. ACM Press.

Gal, Y. and A. Pfeffer (2003). A language for opponent modeling in repeated games. In S. Parsons and P. Gmytrasiewicz (Eds.), *Proceedings of the 5th Workshop on Game Theoretic and Decision Theoretic Agents (GTDT-2003)*.

Greenwald, A. (2003). The 2002 trading agent competition: An overview of agent strategies. *AI Magazine 24*(1), 83–91.

Harel, D. and A. Naamad (1996, October). The STATEMATE semantics of statecharts. *ACM transactions on software engineering and methodology 5*(4), 293–333.

Harsanyi, J. C. (1956). Approaches to the bargaining problem before and after the theory of games: a critical discussion of Zeuthen's, Hicks', and Nash's theories. *Econometrica 24*, 144–157.

He, M. (2004, October). Personal communications.

He, M. and N. R. Jennings (2003). SouthamptonTAC: An adaptive autonomous trading agent. *ACM Trans on Internet Technology 3*(3), 218–235.

He, M. and N. R. Jennings (2004). Designing a successful trading agent using fuzzy techniques. *IEEE Transaction on Fuzzy Systems 12*(3), 389–0410.

He, M., H. Leung, and N. R. Jennings (2003). A fuzzy logic based bidding strategy for autonomous agents in continuous double auctions. *IEEE Transaction on Knowledge and Data Engineering 15*(6), 1345–1363.

Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Communications of the ACM 44*(4), 35–41.

Jennings, N. R., P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. J. Wooldridge (2001). Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation 10*(2), 199–215.

Jennings, N. R. and M. J. Wooldridge (1998). Applications of intelligent agents. In N. R. Jennings and M. J. Wooldridge (Eds.), *Agent Technology: Foundations, Applications, and Markets*, pp. 3–28. Heidelberg, Germany: Springer Verlag.

Juan, T., A. Pearce, and L. Sterling (2002). ROADMAP: Extending the Gaia methodology for complex open systems. In C. Castelfranchi and W. L. Johnson (Eds.), *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), Bologna, Italy*, New York City NY, USA, pp. 3–10. ACM Press.

Jung, H., M. Tambe, and S. Kulkarni (2001). Argumentation as distributed constraint satisfaction: applications and results. In J. P. Müller, E. Andre, S. Sen, and C. Frasson (Eds.), *Proceedings of the Fifth International Conference on Autonomous Agents*, Montreal, Canada, pp. 324–331. ACM Press.

Karunatillake, N. C. and N. R. Jennings (2004). Is it worth arguing? In I. Rahwan, P. Moraitis, and C. Reed (Eds.), *Argumentation in Multi-Agent Systems, First International Workshop, ArgMAS 2004, New York, NY, USA, July 19, 2004, Revised Selected and Invited Papers*, Volume 3366 of *Lecture Notes in Computer Science*, pp. 234–250. Heidelberg, Germany: Springer Verlag.

Kowalczyk, R. and V. Bui (2001). On constraint-based reasoning in e-negotiation agents. In F. Dighum and U. Cortés (Eds.), *Agent-Mediated Electronic Commerce III*, Volume 2003 of *Lecture Notes in Computer Science*, pp. 31–46. Berlin, Germany: Springer Verlag.

Kraus, S. (2001). *Strategic Negotiation in Multi-Agent Environments.* Cambridge MA, USA: MIT Press.

Kraus, S., K. Sycara, and A. Evenchik (1998). Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence 104* (1–2), 1–69.

Kumar, S., M. J. Huber, and P. R. Cohen (2002). Representing and executing protocols as joint actions. In C. Castelfranchi and W. L. Johnson (Eds.), *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), Bologna, Italy*, New York City NY, USA, pp. 543–550. ACM Press.

Kungas, P. and J. Rao (2004). Symbolic agent negotiation for semantic web service exploitation. In Q. Li, G. Wang, and L. Feng (Eds.), *Advances in Web-Age Information Management, 5th International Conference, WAIM 2004, Dalian, China, July 15-17, 2004*, Volume 3129 of *Lecture Notes in Computer Science*. Berlin, Germany: Springer Verlag.

Larson, K. and T. Sandholm (2001). Bargaining with limited computation: deliberation equilibrium. *Artificial Intelligence 132*, 183–217.

Lewicki, R. J., D. M. Saunders, and J. W. Minton (Eds.) (2003). *Essentials of Negotiation.* Boston, MA: McGraw-Hill.

Lilien, G. L., P. Kotler, and S. K. Moorthy (1992). *Marketing Models.* USA: Prentice-Hall Press.

Luck, M., P. McBurney, and C. Preist (2003). *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing).* AgentLink.

Maudet, N. and B. Chaib-draa (2003). Commitment-based and dialogue-game based protocols – new trends in agent communication language. *Knowledge Engineering Review 17* (2), 157–179.

McKeown, K. R. (1985). *Text generation: using discourse strategies and focus constraints to generate natural language text.* Cambridge University Press.

Milch, B. and D. Koller (2000, July). Probabilistic models for agents' beliefs and decisions. In C. Boutilier and M. Goldszmidt (Eds.), *Proceedings of the 16th Annual Conference on Uncertainty in AI (UAI-2000)*, Stanford, California, pp. 389–396. Morgan Kaufmann.

Padgham, L. and M. Winikoff (2004). *Developing Intelligent Agent Systems: A Practical Guide.* John Wiley and Sons.

Parsons, S., C. Sierra, and N. Jennings (1998). Agents that reason and negotiate by arguing. *Journal of Logic and Computation 8*(3), 261–292.

Parsons, S., M. J. Wooldridge, and L. Amgoud (2002). An analysis of formal inter-agent dialogues. In C. Castelfranchi and L. Johnson (Eds.), *Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS-2002)*, New York, USA, pp. 394–401. ACM Press.

Parunak, H. V. D. P. (1999). Industrial and practical applications of DAI. In G. Weiss (Ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pp. 377–421. Cambridge MA, USA: MIT Press.

Rahwan, I. (2004). *Interest-based Negotiation in Multi-Agent Systems.* Ph. D. thesis, Department of Information Systems, University of Melbourne, Melbourne, Australia.

Rahwan, I., P. McBurney, and L. Sonenberg (2003). Towards a theory of negotiation strategy (a preliminary report). In S. Parsons and P. Gmytrasiewicz (Eds.), *Proceedings of the 5th Workshop on Game Theoretic and Decision Theoretic Agents (GTDT 2003)*, pp. 73–80.

Rahwan, I., S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg (2003). Argumentation based negotiation. *Knowledge Engineering Review 18*(4), 343–375.

Rahwan, I., L. Sonenberg, and F. Dignum (2003). Towards interest-based negotiation. In J. Rosenschein, T. Sandholm, M. J. Wooldridge, and M. Yokoo (Eds.), *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, pp. 773–780. ACM Press.

Ramchurn, S. D., N. R. Jennings, and C. Sierra (2003). Persuasive negotiation for autonomous agents: a rhetorical approach. In C. Reed, F. Grasso, and G. Carenini (Eds.), *Proceedings of the IJCAI Workshop on Computational Models of Natural Argument*, pp. 9–17. AAAI Press.

Roberts, J. H. and J. M. Lattin (1991). Development and testing of a model of consideration set composition. *Journal of Marketing Research 28*(4), 429–440.

Rosenschein, J. and G. Zlotkin (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers.* Cambridge MA, USA: MIT Press.

Sadri, F., F. Toni, and P. Torroni (2001). Logic agents, dialogues and negotiation: an abductive approach. In K. Stathis and M. Schroeder (Eds.), *Proceedings of the AISB 2001 Symposium on Information Agents for E-Commerce.*

Sadri, F., F. Toni, and P. Torroni (2002). Abductive logic programming architecture for negotiating agents. In *Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA-2002)*, Volume 2424 of *Lecture Notes in Computer Science*, pp. 419–431. Berlin, Germany: Springer Verlag.

Sandholm, T. (2002). eMediator: A next generation electronic commerce server. *Computational Intelligence, Special issue on Agent Technology for Electronic Commerce 18*(4), 656–676.

Sandholm, T. and V. Lesser (2002). Leveled commitment contracting: A backtracking instrument for multiagent systems. *Artificial Intelligence Magazine 23*(3), 89–100.

Schut, M. and M. Wooldridge (2001). The control of reasoning in resource-bounded agents. *Knowledge Engineering Review 16*(3), 215–240.

Sierra, C., N. R. Jennings, P. Noriega, and S. Parsons (1998). A framework for argumentation-based negotiation. In M. Singh, A. Rao, and M. J. Wooldridge (Eds.), *Intelligent Agent IV: 4th International Workshop on Agent Theories, Architectures and Languages (ATAL-1997)*, Volume 1365 of *Lecture Notes in Artificial Intelligence*, pp. 177–192. Berlin, Germany: Springer Verlag.

TAC (2003). *The Trading Agent Competition.* http://www.sics.se/tac/: World Wide Web.

Tambe, M., E. Bowring, H. Jung, G. Kaminka, R. Maheswaran, J. Marecki, P. J. Modi, R. Nair, S. Okamoto, J. P. Pearce, P. Paruchuri, D. Pynadath, P. Scerri, N. Schurr, and P. Varakantham (2005). Conflicts in teamwork: hybrids to the rescue. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, New York, NY, USA, pp. 3–10. ACM Press.

Torroni, P. (2002). A study on the termination of negotiation dialogues. In C. Castelfranchi and L. Johnson (Eds.), *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, New York, USA, pp. 1223–1230. ACM Press.

von Stengel, B. (2002). Computing equilibria for two-person games. In R. Aumann and S. Hart (Eds.), *Handbook of game theory*, Volume 3, Chapter 45, pp. 1723–1759. Amsterdam, Netherlands: Elsevier Science Publishers (North-Holland).

Walton, D. N. and E. C. W. Krabbe (1995). *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning.* Albany NY, USA: SUNY Press.

Wooldridge, M., N. R. Jennings, and D. Kinny (2000). The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems 3*(3), 285–312.

Wooldridge, M. J. (2002). *An Introduction to MultiAgent Systems.* Chichester, England: John Wiley & Sons.

Young, M. A. (2001). *Rational Games: A Philosophy of Business Negotiation from Practical Reason.* Westport, CT, USA: Quorum Books.

Zambonelli, F., N. R. Jennings, and M. Wooldridge (2003). Developing multiagent systems: the Gaia methodology. *ACM Trans on Software Engineering and Methodology 12*(3), 317–370.

List of Figures

List of Tables